

"Express Mail" mailing label number:

EV335379729US

**METHOD AND SOFTWARE FOR IMPROVED CIRCUIT PATH TIMING
PREDICTIONS, AND CIRCUIT DESIGN PRODUCED THEREWITH**

Aveek Sarkar
Shian-Jiun Fu
Peter Lai
Rambabu Pyapali

FIELD OF THE INVENTION

[1001] The invention relates generally to analyzing circuit timing, and more particularly to improving the accuracy and/or speed of circuit path timing predictions.

BACKGROUND

[1002] As the speed at which processors and other circuits operate has increased, the focus of many circuit design techniques has shifted towards ensuring compliance with stringent timing requirements. To make sure that circuits they design conform to these stringent timing requirements, circuit designers use a number of different timing prediction/modeling tools. Many of these tools generate timing models of a circuit using pre-constructed models of common elements or groups of elements stored in library files. Many of these tools also use simplified equations based on generalized assumptions. Because of the way in which these modeling tools operate, the circuit timing estimates they generate are often not completely accurate.

[1003] Other, more accurate modeling tools are available. Their accuracy, however, comes at the expense of speed. In order to achieve greater accuracy, these more accurate modeling tools generally employ more complex equations, and/or take into account a larger universe of variables in generating timing estimates. Unfortunately, when dealing with modern circuits having millions of transistors, these more accurate tools are too slow to use for analyzing all of the signal paths in a circuit.

[1004] Consequently, circuit designers are often placed in the unenviable position of choosing between using fast modeling tools providing relatively inaccurate results, and tools that are significantly slower but provide more accurate results. It should be apparent, therefore, that an improved way of modeling the timing of complex circuits would be advantageous.

SUMMARY

[1005] The timing response of a circuit path is predicted by modeling the circuit path using two different timing models. The variation between the timing responses, produced by each of the timing models is used to generate a correction factor, which is then applied to one of the timing models. Once the correction factor has been applied to a timing model, the model is used to produce a corrected timing prediction for the modeled circuit path.

[1006] The correction factor can be used to correct coarse timing predictions for a large number of circuit paths in a relatively short period. For example, one of the two different timing models may be used to obtain coarse timing response estimates for all of the circuit paths in a particular design block. The other timing model may be used to obtain a refined timing estimate of one or more selected circuit paths from the same design block. The coarse estimate and the refined estimate for the selected circuit paths are compared, and a correction factor is generated. The correction factor can then be used to adjust the coarse timing estimates of all the circuit paths – not just the selected circuit paths.

[1007] In most cases, the coarse estimate will be less accurate than the refined estimate, because the model generating the coarse estimate makes more modeling assumptions, use pre-calculated library estimates, etc. For the same reasons, however, it will generally be quicker to obtain the coarse timing estimates than to obtain the refined timing estimates. Consequently, by generating a correction factor based on variations between the coarse timing estimates and the refined timing estimates, and by adjusting the coarse timing estimates based on the correction factor, the accuracy of the coarse timing estimates can be improved without requiring the timing of each circuit path to be predicted using the slower, but more accurate, refined timing model.

[1008] The improved method of predicting the timing response of a circuit path described above may be implemented as a program of instructions encoded into a computer readable medium to be executed by a data processor. Such a program of instructions may be used as part of a program specifically designed to implement the teachings set forth herein, or in conjunction with other modeling tools commonly available, such as Pathmill®, Dynacore®, and Simulation Program for Integrated Circuits Emphasis (SPICE).

[1009] The method and/or software described in the present disclosure may also be used to produce a computer readable medium encoding an integrated circuit design. Such a computer readable medium may be useful for transferring circuit design information to semiconductor foundries, for transferring information between design groups, and the like.

BRIEF DESCRIPTION OF THE DRAWINGS

[1010] The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

[1011] **FIG. 1** is a block diagram illustrating processing elements used to generate timing estimates for substantially all timing paths through a circuit block, according to one embodiment of the invention;

[1012] **FIG. 2** is a block diagram illustrating a custom designed circuit block having multiple timing paths, according to one embodiment of the invention;

[1013] **FIG. 3** is a chart showing the application of a correction factor, generated using selected timing paths in a circuit block, to multiple timing paths in the circuit block according to one embodiment of the invention;

[1014] **FIG. 4** is a flow chart illustrating a method according to one embodiment of the invention; and

[1015] FIG. 5 is a block diagram of a processing system suitable for reading and/or executing a computer readable medium encoded with an integrated circuit design or a program of instructions.

[1016] The use of the same reference symbols in different drawings indicates similar or identical items.

DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

[1017] FIG. 1 illustrates a system 100 according to an embodiment of the present invention. System 100 includes various computational or processing elements that may be used to generate an improved timing estimate for many timing paths in a custom circuit block by using a few selected paths to generate a correction factor, and then applying that correction factor to all of the timing paths, including the unselected timing paths. The elements illustrated in FIG. 1 include Coarse Model element 110 and Refined Model element 120 for modeling timing paths, Correction Factor Generator 130 for generating a correction factor 180 indicating a variance between the timing estimates generated by Coarse Model element 110 and Refined Model element 120, and Correction Application element 140 for applying the correction factor 180 to timing estimates generated by Coarse Model element 110.

[1018] In one form, system 100 receives a description of a custom circuit design block, such as Circuit Block Netlist 150, at Coarse Model element 110. Circuit Block Netlist 150 may be a node-level description of a circuit to be modeled, a component level description, or some other type of circuit description suitable for use by Coarse Model element 110. Circuit Block Netlist 150 may include a description of an entire electronic circuit, a specific functional circuit block, or any other portion of an electronic circuit to be modeled. In many cases, the circuit described by Circuit Block Netlist 150 will include hundreds or even thousands of possible timing paths, with each path representing a series of components or blocks that an electronic signal must travel to move from an input of the circuit to an output of the circuit.

[1019] Coarse Model element 110 identifies the timing paths in the circuit block described by Circuit Block Netlist 150, and generates Coarse Timing Estimates 160 for all or substantially all identified timing paths. Coarse Timing Estimates 160 may

include estimates of signal propagation times, estimates of signal propagation delays, and the like. To generate timing estimates quickly, Coarse Model element 110 employs a coarse timing model that incorporates the use of simplified timing assumptions and/or pre calculated values obtained from a library/database of known elements. For example, in modeling the timing response of a transistor, Coarse Model element 110 may disregard certain capacitance values, such as Miller capacitance, and second order effects such as non-linearity of the rise and fall times of the input signals. Alternatively, Coarse Model element 110 may use a time estimate calculated in advance for an average transistor, or extrapolate a time estimate from a lookup table.

[1020] In at least one embodiment, Coarse Model element 110 also generates a node-level description of selected timing paths, such as Selected Path Netlist 152, and provides the Selected Path Netlist 152 to Refined Model element 120. It should be noted, however, that Selected Path Netlist 152 may need to be verified before being provided to Refined Model element 120, due to possible errors that may be introduced in the modeling process by Coarse Model element 110. For example, Coarse Model element 110 may ignore certain devices described in the original netlist, Circuit Block Netlist 150, because delays introduced by the ignored devices are assumed by Coarse Model element 110 to be inconsequential.

[1021] In other embodiments, however, Coarse Model element 110 does not generate Selected Path Netlist 152, or if Coarse Model element 110 does generate a netlist, the netlist is not provided to Refined Model element 120. In such an embodiment, portions of Circuit Block Netlist 150 defining selected paths may be delivered directly to Refined Model element 120 upon receipt by System 100. Despite the possible inaccuracies of netlists generated by Coarse Model element 110, however, it may be more efficient to have Coarse Model element 110 to generate Selected Path Netlist 152, and then verify any suspect timing paths, than to construct Selected Path Netlist 152 by hand or otherwise.

[1022] Selected Path Netlist 152 is, in at least one embodiment, a node-level netlist describing particular timing paths selected from the universe of timing paths described in Circuit Block Netlist 150. The timing paths included in Selected Path

Netlist 152 may be chosen manually, chosen by Coarse Model element 110, or otherwise. The process by which timing paths are selected for inclusion in Circuit Block Netlist 150 may be random, pseudo random, based on maximum or minimum timing requirements, based on the number of circuit elements in a path, based on circuit layout requirements, based on a desired circuit characteristic, or the like.

[1023] Refined Model element 120 receives Selected Path Netlist 152 from Coarse Model element 110, and generates refined timing estimates 170 for each of the paths described in Selected Path Netlist 152. Refined timing estimates 170 may include estimates of signal propagation times, estimates of signal propagation delays, and the like. Refined Model element 120 generates more accurate timing estimates than the timing estimates generated by Coarse Model element 110, because Refined Model element 120 makes fewer simplifying assumptions, and does not rely on precalculated library values. In addition, Refined Model element 120 may also take into account secondary effects such as Miller capacitances, certain parameters used in SPICE models, and multi-leg switching scenarios, and/or use otherwise more precise modeling algorithms.

[1024] Coarse Model element 110 generates course timing estimates for all paths 160. Included in these coarse timing estimates, are coarse timing estimates for the selected paths 162. It should also be noted that, although in the illustrated embodiment Coarse Model element 110 initially generates coarse timing estimates for all paths 160, and the coarse timing estimates for selected paths 162 are chosen from among the coarse timing estimates for all paths 160, in other embodiments Coarse Model element 110 initially generates only the coarse timing estimates for the selected paths 162 needed by Correction Factor Generator 130. Any remaining coarse timing estimates may then be generated after correction factor 180 has been generated.

[1025] The coarse timing estimates for the selected paths 162 and the refined timing estimates for the selected paths 170 are both provided to correction factor generator 130. Correction Factor Generator 130 uses the coarse timing estimates for selected paths 162 and the refined timing estimates for selected paths 170 to determine any variation between the coarse and refined timing estimates for each

selected path, and then to generate a correction factor 180 based on that variation. The way in which Correction Factor Generator 130 generates correction factor 180 will be discussed in greater detail with reference to FIG. 3.

[1026] Once correction factor 180 has been generated, Correction Application element 140 uses correction factor 180 to generate a corrected coarse response for all paths 190. Correction Application element 140 adjusts the coarse timing estimates for all paths 160 based on correction factor 180. For example, if correction factor 180 is a simple offset, then Correction Application element 140 adds the offset to the coarse timing estimate to produce corrected timing responses. If correction factor 180 is a scaling factor, then Correction Application element 140 multiplies the coarse timing estimates by the correction factor. If correction factor 180 is a more complex statistical function, then Correction Application element 140 applies the function to the coarse timing estimates.

[1027] It will be appreciated that although FIG. 1 illustrates Correction Application Element 140 as physically separate from Coarse Model element 110, in other embodiments Coarse Model element 110 includes hardware and/or software components for implementing the functions of Correction Application Element 140. In such a case, correction factor 180 is applied to the Coarse Model element 110, and corrected coarse responses are provided directly from Coarse Model 110.

[1028] Referring next to FIG. 2, a block diagram of a circuit having multiple timing paths is illustrated, and designated generally as circuit block 200. Circuit block 200 includes eight inputs 201-208, output 209, and various circuit elements 210-250. Any signal traveling from one of the inputs 201-208 to output 209 moves along a timing path. For example, a signal traveling from input 201 to output 209 travels along path 260, a signal traveling from input 202 travels along path 262, and a signal traveling from input 203 to output 209 travels along path 264. The Coarse Model element 110 (FIG. 1) and Refined Model element 120 (FIG. 1) operate on netlists, or other similar descriptions of circuit block 200 familiar to those skilled in the art, to generate either coarse or refined estimates, respectively, of how long it will take a signal to travel along a particular path such as paths 260, 262, or 264. It will be appreciated that circuit block 200 is highly simplified, and that circuit blocks

typically have multiple outputs as well as multiple inputs, so that a particular input may have paths to multiple outputs, a single input may have multiple paths to a single output, etc. For ease of discussion, however, each input 201-208 is constrained to have only a single path to output 209.

[1029] Referring now to FIGS. 2 and 3, a method according to an embodiment of the present invention will be discussed. Chart 300 includes input column 310, coarse estimated timing column 320, refined estimated timing column 330, variance column 340, correction factor column 350 and corrected column 360. Input column 310 includes input designators 1-8, which represent inputs 201-208, respectively, of circuit block 200. As noted above, to simplify discussion each input 201-208 of illustrated circuit block 200 has only one path to output 209. Consequently, rows 1-8 each include information associated with a respective path for which timing is to be estimated. For example, row 1 includes information for path 260, which is the path from input 201 to output 209. Likewise, row 2 includes information for path 262, and row 3 includes information for path 264. The remaining rows 4-8 include information for paths beginning at input 204-208, respectively. Note that rows 1-3 are designated as selected 370, and rows 4-8 are designated as unselected 380. The rows designated as selected 370 include information for paths chosen to have both coarse and refined timing estimates generated.

[1030] A method according to one embodiment begins with Coarse Model element 110 (FIG. 1) generating coarse timing estimates for all selected paths 370 and unselected paths 380. Exemplary values for the coarse timing estimates are listed in column 320. No units are included chart 300, but those skilled in the art will recognize that the values may be given in milliseconds, microseconds, or any other suitable unit of time, depending on the speed of the circuit being analyzed. Refined timing estimates are generated for each of the selected paths 370. Column 330 shows exemplary values for each of the selected timing paths.

[1031] Note that no refined timing estimate is performed on the majority of the timing paths, while coarse timing estimates are performed on all of the timing paths. Each coarse timing estimate can normally be obtained much quicker than a corresponding refined timing estimate. For example, in many cases a coarse timing

estimate can be obtained for all paths of a complicated circuit in a matter of hours, while performing a refined estimate for the same paths might take a matter of days. Consequently, the time savings obtained by performing timing analysis using the methods described herein can be significant.

[1032] Some or all of the refined timing estimates in column 330 may or may not be the same as the coarse timing estimates in column 320, depending on the circuit being evaluated, the accuracy of coarse timing model, etc. Generally, however, at least some of the refined timing estimates are different from the coarse timing estimates. These differences are illustrated by the values in variance column 340. In the present example, the variance is illustrated as a difference between the coarse and refined timing estimates. This variance is then used to generate a correction factor, examples of which are shown in column 350.

[1033] The Correction Factor Generator (FIG. 1) generates the correction factor, shown in Column 350, based on the variances in column 340. In the illustrated example, the correction factor is the average variance between the coarse and refined estimated responses for all of the selected paths 370. The variance of path 260, shown in row 1, is -3, since the coarse timing estimate was 89 and the refined timing estimate was 86. The variance in path 262, shown in row 2, is -2, since the coarse timing estimate was 86 and the refined timing estimate was 84. The variance in path 264, shown in row 3, is -4. The average of these three variances (-3, -2, and -4) is -3. This average variance is chosen as a correction factor, and then applied to all of the timing paths, such that the corrected estimated timing shown in column 360 represents the coarse timing in column 320 adjusted by the correction factor shown in column 350. As a result, the corrected timing estimate for path 260 is 86, the corrected timing estimate for path 262 is 83, and the corrected timing estimate for path 264 is 91. The correction factor is likewise applied to the unselected timing paths 380, so that the relatively large number of timing paths that were not evaluated using the refined timing model are adjusted by an amount determined by the variance between the coarse and refined timing estimates of a relatively smaller number of timing paths.

[1034] Note that although the correction factor discussed in relation to FIG. 3 is a simple average of the variances, other suitable correction factors may be used. For example, the mean of the variances of all selected paths may be used as a correction factor. Alternatively, the correction factor may be a more complex function related to the number of circuit elements in a particular path. For example, assume that three paths were analyzed, with variances of +1, +2 and +3, respectively. Assume furthermore that the first path includes 10 elements, the second path includes 20 elements, and the third path includes 30 elements. In such a case, an average correction factor of +2 might be applied, or a correction factor of +1 for each 10 elements might be applied. In that case, a different correction factor might be applied to a particular path depending on how many circuit elements were in the path. For example, any path that included 40 elements would have a +4 correction applied, while a path including only 5 elements would have a +0.5 correction applied. Similarly, other functions including offsets, multiplication factors, or any suitable mathematical formula may be used as a correction factor.

[1035] In at least one embodiment, a statistical analysis is performed on the variances of all selected timing paths. Such a statistical analysis can include a determination of a coefficient of variation, so that if the coarse-refined timing difference, i.e. the statistical variation (variance) of a particular path, is highly positive for some paths and highly negative for others, then one or more timing paths may be flagged for verification by hand or otherwise. For example, a statistical variation may be determined by first generating a correction factor for selected paths in a design block. Next, the standard deviation and mean of the correction factors are determined, and a ratio of the standard deviation to the mean is calculated. When the ratio exceeds a certain value, for example 20% percent, each of the correction factors will be verified. Statistical analysis may also be used to help select the proper number of paths for both coarse and refined timing analysis. For example, if there are 10,000 possible paths, a statistically significant number of paths may be selected to improve the likelihood that any correction factor generated by Correction Factor Generator 130 (FIG. 1) will be accurate.

[1036] Referring next to FIGS.1 and 4, a flowchart describing a method according to an embodiment of the present invention will be discussed, and is designated

generally as method 400. The method 400 begins at step 410, when a circuit block design is provided to Coarse Model element 110. The circuit block design may be provided in the form of a netlist, or a schematic or other representation of the circuit design may be delivered to Coarse Model element 110. In the case where the circuit design is provided to Coarse Model element 110 in a schematic form, Coarse Model element 110 extracts a netlist or some other suitable description of the circuit. Various ways of providing a circuit description to Coarse Model 110 are known to those skilled in the art.

[1037] The method proceeds to step 420, in which Coarse Model element 110 generates one or more coarse timing estimates. In at least one embodiment, Coarse Model element 110 generates coarse timing estimates for all of the paths in the circuit, but in other embodiments Coarse Model element 110 generates coarse timing estimates for only a selected portion of the circuit paths. If Coarse Model element 110 generates coarse timing estimates for all the circuit paths, step 430 is performed to select some of the circuit paths for further analysis by Refined Model element 120. The circuit paths may be selected randomly, by use of a statistical formula, based on a maximum or minimum timing criterion, or by any other suitable method.

[1038] In step 440, Refined Model 120 generates refined timing estimates for the paths selected in step 430. Once the selected paths have been analyzed by both Coarse Model element 110 and Refined Model element 120, step 450 is performed, in which Correction Factor Generator 130 generates a correction factor based on the variance between the refined timing estimate and the coarse timing estimate for each selected path. Finally, the correction factor is applied to the coarse timing model and/or the coarse timing estimates are adjusted based on the correction factor.

[1039] The method illustrated in FIG. 4, as well as the methods discussed earlier with reference to FIGS. 1-3, can be implemented as a program of instructions executed using a data processor. For example, each of the major elements shown in FIG. 1 may be implemented as a subroutine, a stand alone program, or the like. In at least one embodiment, off-the shelf timing analysis tools are used to implement Coarse Model element 110 and Refined Model element 120. The output of these off-the-shelf tools is then used in combination with software subroutines, programs, etc. used

to implement Correction Factor Generator 130 and Correction Application element 140.

[1040] In one form, the teachings set forth herein may be used to produce a computer readable medium product that encoded with a circuit design. Such a circuit design may be provided, in the form of a computer readable medium product, to foundries or other design facilities, for prototyping, manufacturing, further design, etc. The computer readable medium product may include a random access memory, a read only memory, a magnetic tape, a magnetically encodable disk, an optically encodable tape, an optically encodable disk, or a propagated signal.

[1041] Referring next to FIG. 5, a processor suitable for use with a computer readable medium encoding a circuit design and for use in implementing a program of instructions stored on a computer readable medium is discussed. As will be appreciated, the term “computer readable medium” encompasses memory devices such as various forms of computer memory, including for example, random access memory (RAM) and read only memory (ROM), electromagnetic disk drives, optical drives, various forms of propagated signals, magnetic tapes, optical tapes, paper tapes, and any similar medium used in transporting or storing computer readable information.

[1042] The data processor 500 illustrated in FIG. 5 includes a number of devices connected to a common system bus 510. Processor unit 520 includes one or more devices, commonly referred to as central processing units (CPU) that obtain information to be processed over system bus 510, process the information, and output results to one of the connected devices. RAM 530 and ROM 540 store programs and/or data for execution by processor unit 520. Removable storage 560 may include storage devices having removable media, such as floppy disks, compact disks, tape drives, memory sticks, and the like. Fixed storage 570 includes such devices as hard disk drives and the like that do not have removable media. Generally, both removable storage 560 and fixed storage 570 are used for bulk storage of information and/or programs. The information and programs stored fixed storage 570 and removable storage 560 are usually copied to RAM 530 for faster access by processor unit 520. Finally, communications device 550 includes network interface cards,

modulator/demodulators (modems), and the like, while input output unit (I/O) 580 includes devices such as keyboards, speakers, monitors, and pointing devices.

[1043] Although an embodiment of the present invention has been shown and described in detail herein, along with certain variants thereof, many other varied embodiments that incorporate the teachings of the invention may be easily constructed by those skilled in the art. Accordingly, the present invention is not intended to be limited to the specific form set forth herein, but on the contrary, it is intended to cover such alternatives, modifications, and equivalents, as can be reasonably included within the spirit and scope of the invention.